I confirm that the work of this assignment is completely my own. By turning in this assignment, I declare that I did not receive unauthorized assistance. Moreover, all deliverables including, but not limited to the source code, lab report and output files were written and produced by me alone.

**1. Program Explanation**

The assignment is to build on the functionality of PA#2 while adding one additional BankStatements class. I modified the functionality of creating the hashtable containing all the customers and writing to the new and updated csv file by making them dynamic. That it, making sure the methods will work even is the csv columns are in different order. I also added the bank manager ability to add a new user and generate a bank statement for a customer. The customer can now also make a new account for themselves. I tackled this problem by taking it one piece at a time and spacing the pieces out throughout the weeks. I first made sure I had a very good Idea of how I wanted to execute each functionality. Thinking through exactly how the classes and methods would work with one another was very helpful since it prevented me from starting to execute one idea only to realize it is not possible.

**2. What did I learn?**

a result of this assignment, I learned how beneficial it is to plan out the structure of the program first before going into coding. I did this in both PA1 and PA2 but I have noticed these skills get better with more practice. I now feel more confident in correctly executing an idea I have. I also got a better understanding of coding different relationships between classes. My solution can be improved by getting rid of some repeating code such as verifying is a user is in the hashTable among other things. As of now it is hardcoded in one aspects. It only works if the columns for the transaction actions are in a specific way and I can correct that in the next PA. One approach of improvement I have is to include the account numbers inside of the Hashtable key. This way I could access a customer object by knowing an account number. this would be helpful int the bank manager functionality. This programming assignment took me around 7-9 hours to complete throughout the 3 weeks.

**3. Solution Design**

In this programming assignment, I created the functionality for the user to create a new account. I did this in the main by simply asking if they would like a new checking/savings/credit account, only if they did not already have one and asking to the necessary arguments. I also allowed the bank manager to create a new user. The assumption I made was that in order to become a user, one must open a saving account. I also did this in the main by asking the user for the necessary arguments. I read the BankUsers csv file now as well. My approach was to first get the important indices such s name and account balances. And then once the customer object is created, I use a switch case for find the correct fields in the file and match them to the corresponding attributes. I also added the Bank statement class which generates a Bank statement for a specific person. My approach was creating an arrayList of transactions, which is attribute of Account. Each time the balance of an account changes, I add a transaction string which describes what happened to the arrayList. In the BankStatement Class, customer is an attribute so I have access to each arraylist and can write them to a txt file if bank statement is requested.

**4. Testing**

I tested my program using black-box and white-box testing. I used both to ensure that my code was running as expected and that it made sense to the user who knows nothing about the inner workings of the program. My testing practices can be improved by testing more often because When I wait too long in between tasting, the problems get bigger. I tested by running the program and inputting things myself and by testing some methods using junit.

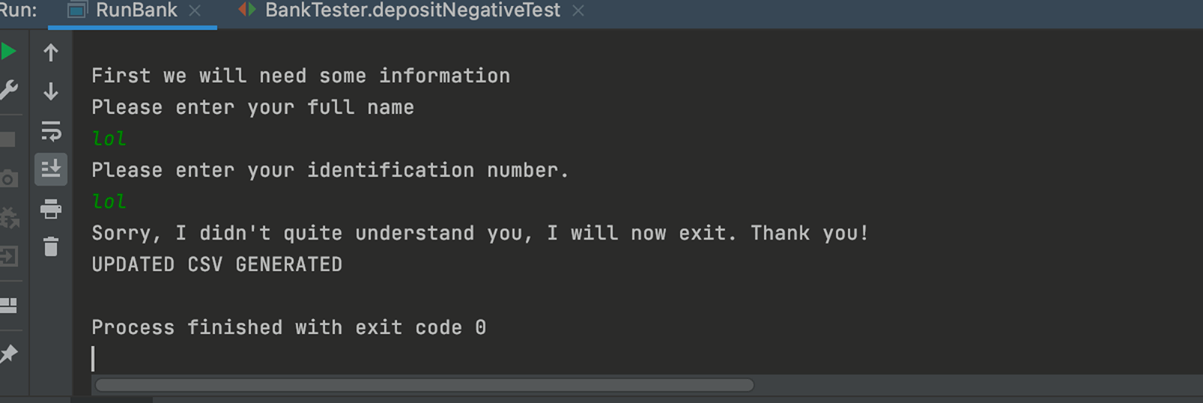
The latest test I did are below. The log.txt file generated by my program and the BankUsersUpdated.csv file reflect this testing.

**5. Test results**

How does she handle input mismatch?

I tested this by inputting the incorrect format of information into the console. This is black box testing since I did not know how Nadia coded these exception handlings. When inputting random things. I did not see any exceptions thrown by java, Only Nadia’s message indicting that the program could not understand my input

This is the screenshot from Nadia’s code handling mismatch exceptions well.



Do transaction actions account for errors? Wil it go on to next transaction action is one does not work?

I tested this by making sore an error would occur in the transaction actions(e.g. an account going bellow 0) and running transaction actions and seeing is all actions in the file are run or of it stops at some point because of one error. This is a combination of white box and back box since I know how the code will most probably break since I did the project myself, but black box since I did not see Nadia’s code and did not know for sure how it would respond to my attempts of breaking it. When running it I noticed not all of the transaction actions were logged, so there was some issue there and it did not work as expected.

**6. Test results**

These are screenshots of what I inputted in the console and the corresponding logs the program produced.

